UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/975,286 | 10/10/2001 | Christopher Peiffer | 1014-152US01 | 9849 |

72689 7590 03/10/2010

SHUMAKER & SIEFFERT, P.A
1625 RADIO DRIVE , SUITE 300
WOODBURY, MN 55125

| EXAMINER | |
|---|---|
| PATEL, HARESH N | |
| ART UNIT | PAPER NUMBER |
| 2454 | |

| NOTIFICATION DATE | DELIVERY MODE |
|---|---|
| 03/10/2010 | ELECTRONIC |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

pairdocketing@ssiplaw.com

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 09/975,286 | PEIFFER, CHRISTOPHER | |
| | Examiner | Art Unit | |
| | HARESH N. PATEL | 2454 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *18 February 2010*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-5,7,8 and 11-26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5,7,8 and 11-26* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

## DETAILED ACTION

1. Claims 1-5, 7, 8, 11-26 are subject to examination. Claims 6, 9, and 10 are cancelled.

2. The finality of previous office action is withdrawn and the prosecution is hereby reopened. This office action is made non-final considering new grounds of rejection, which were not present in previous office action.

### *Claim Rejections - 35 USC § 101*

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claim 26 is rejected under 35 U.S.C. 101 because the claimed invention is directed to a non-statutory subject matter. The claim 26 contain medium which is not limited to hardware. Replacement of "computer storage medium" with --non-transitory computer storage medium-- is suggested to overcome 35 U.S.C. 101 rejections.

The United States Patent and Trademark Office (USPTO) is obliged to give claims their broadest reasonable interpretation consistent with the specification during proceedings before the USPTO. *See In re Zletz*, 893 F.2d 319 (Fed. Cir. 1989) (during patent examination the pending claims must be interpreted as broadly as their terms reasonably allow). The broadest reasonable interpretation of a claim drawn to a computer readable medium (also called machine readable medium and other such variations) typically covers forms of non-transitory tangible media and transitory propagating signals *per se* in view of the ordinary and customary meaning of computer readable media, particularly when the specification is silent. *See* MPEP 2111.01. When the broadest reasonable interpretation of a claim covers a signal *per se*, the claim must be rejected under 35 U.S.C. § 101 as covering non-statutory subject matter. *See In re Nuijten*, 500 F.3d 1346, 1356-57 (Fed. Cir. 2007) (transitory embodiments are not directed to statutory subject matter) and *Interim Examination Instructions for Evaluating Subject Matter Eligibility Under 35 U.S.C. § 101*, Aug. 24, 2009; p. 2.

The USPTO recognizes that applicants may have claims directed to computer readable media that cover signals *per se*, which the USPTO must reject under 35 U.S.C. § 101 as covering both non-statutory subject matter and statutory subject matter. In an effort to assist the patent community in overcoming a rejection or potential rejection under 35 U.S.C. § 101 in this situation, the USPTO suggests the following approach. A claim drawn to such a computer readable medium that covers both transitory and non-transitory embodiments may be amended to narrow the claim to cover only statutory embodiments to avoid a rejection under 35 U.S.C. § 101 by adding the limitation "non-transitory" to the claim. *Cf. Animals – Patentability*, 1077 *Off. Gaz. Pat. Office* 24 (April 21, 1987) (suggesting that applicants add the limitation "non-human" to a claim covering a multi-cellular organism to avoid a rejection under 35 U.S.C. § 101). Such an amendment would typically not raise the issue of new matter, even when the specification is silent because the broadest reasonable interpretation relies on the ordinary and customary meaning that includes signals *per se*. The limited situations in which such an amendment could raise issues of new matter occur, for example, when the specification does not support a non-transitory embodiment because a signal *per se* is the only viable embodiment such that the amended claim is impermissibly broadened beyond the supporting disclosure. *See, e.g., Gentry Gallery, Inc. v. Berkline Corp.*, 134 F.3d 1473 (Fed. Cir. 1998).

## *Claim Rejections - 35 USC § 112*

The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it

pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

4. Claims 1-5, 7, 8, 11-26 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claims 1 and 26 contain operation between different length of **segments** and/or unknown string (note: **as claimed**, the claimed subject matter is not limited to same length of segments and/or unknown string), which is not described in the specification of this application under examination in such a way as to reasonably convey to one skilled in the relevant art to use and/or make the invention. Note: **bitwise** exclusive OR operation can only be done on two strings (meaning two inputs of the bitwise exclusive OR) with same length.

The specification of this application also **clearly** states at line 15, page 11- line 6, page 12,Turning to Fig. 6, a string matching method according to one embodiment of the present invention is shown at 100. Typically the **lengths** of the two strings to compare are **known** and **equivalent**. Method 100 includes identifying a predefined string at 102 and identifying an unknown string to compare with the predefined string at 104. The predefined string is typically selected from a record of strings, each having only alphabetic characters. Typically, the predefined string is an HTTP header. The unknown string is typically a header to be compared with the predefined string. Both segments typically contain **the same number of characters**.

***Claim Rejections - 35 USC § 112***

5. Claims 1-5, 7, 8, 11-26 rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential steps, such omission amounting to a gap between the steps. See MPEP § 2172.01. Claims 1 and 26 contain "the indication for the case-insensitive string

match indicates whether **all characters** of the unknown string within the network message match **all corresponding characters** of the selected predefined string". The omitted steps in claims 1 and 26 are: performing operation between an ASCII binary representation of **all characters** of the unknown string and an ASCII binary representation of **all characters** of the selected predefined string. The claims 1 and 26 as claimed, rather all characters of the unknown string and the selected predefined string, has only **a segment** of the unknown string and **a segment** of the selected predefined string for the operation, which is not sufficient. For example, if selected predefined string is "Accept-Encoding" (HTTP header) and unknown string is "Accept-Language"; if all characters are compared than the case-insensitive string match would indicate correct result that " Accept-Encoding" and " Accept-Language" are not same.

However, **as claimed**, if only **a segment** of the predefined string, meaning only "Accept" of "Accept-Encoding" is compared with only "Accept" (segment of the unknown string "Accept-Language") the case-insensitive string match **would indicate incorrect** result. Note: the segment of a predefined string can be only one character or any number of characters, versus the segment of the unknown string can be only one character or any number of characters.

The omitted steps in claims 1 and 24-26 are: performing operation between an ASCII binary representation of **same number of characters** of the unknown string and an ASCII binary representation of the selected predefined string.

Note: claim 25 contains "performing a bitwise exclusive OR operation on binary representations of the client HTTP header and the known HTTP header selected from the database", in which **all characters of the strings** are used for the operation, versus, claims 1-5, 7, 8, 11-24, 26, are not limited to having **all characters of the strings** for the operation and

hence subject to the 35 U.S.C. 112 rejections. The claims are not limited to having **all same number of characters.**

**As claimed**, if the predefined string (longer) "Accept-Encoding" is compared with "Accept" being (shorter) unknown string, the case-insensitive string match **would indicate incorrect** result.

The preamble in claim 1 contains **comparing** an unknown **string** to a predefined **string**. Since the claimed comparison steps utilizing only segments, i.e., "performing a bitwise exclusive OK operation between an ASCII binary representation of **at least a segment** of the unknown string and an ASCII binary representation of **at least a segment** of the selected predefined string", lines 11-13, claim 1; "comparing an unknown string to a predefined string" line 1, claim 1 is not done. Also the indication for the case-insensitive string match indicates whether **all characters of the unknown string** within the network message match all **corresponding characters of the selected predefined string** so as to match one of the known headers of the network communication protocol is not done.

The specification of this application also **clearly** states at line 15, page 11- line 6, page 12,Turning to Fig. 6, a string matching method according to one embodiment of the present invention is shown at 100. Typically the **lengths** of the two strings to compare are **known** and **equivalent**. Method 100 includes identifying a predefined string at 102 and identifying an unknown string to compare with the predefined string at 104. The predefined string is typically selected from a record of strings, each having only alphabetic characters. Typically, the predefined string is an HTTP header. The unknown string is typically a header to be compared with the predefined string. Both segments typically contain **the same number of characters**.

The specification of this application also **clearly** states at line 5, page 9, the unknown string can be **incoming data** from remote client. Since, the incoming data is unknown, the unknown string is **not limited** to the same length of the predefined string or the same length of the predefined flag, etc. Since, the content of the incoming data is unknown, the unknown string is **not limited** to containing ASCII or binary or other type of content.

Providing of two strings with similar length, similar type of data, for example both ASCII rather one ASCII and another of any type, etc before the first bitwise XOR operation is performed is missing.

Claims 1 and 26 contain only below steps for accomplishing case-insensitive string match between two strings in which one is a predefined string and another is an unknown string. As per the claimed subject matter, number of characters, length, etc. of "unknown string" are not known and can be different than the predefined string. Also as claimed, the predefined flag is not limited to ASCII binary representation or a Boolean flag or any character or any thing else. Further, the value predefined flag can be same for all string comparisons (when the same method steps are performed with different "unknown strings"), meaning the predefined flag can be "0" all the time regardless of different "unknown string" and "predefined string". Similarly the predefined flag can be "1" all the time regardless of different "unknown string" and "predefined string". The result of the exclusive OR operation can be of different length as compared to the predefined flag length and hence a bitwise operation between them would not be possible and/or produce incorrect result. Comparing predefined flag, for example, "0", when the same method steps are performed with different "unknown strings", would produce same indication and/or incorrect indication. Comparing predefined flag, for example, "1", when the same method steps are

performed with different "unknown strings", would produce same indication and/or incorrect indication. Comparing predefined flag, for example, "A", when the same method steps are performed with different "unknown strings", would produce same indication and/or incorrect indication. The predefined flag is **not limited** to ASCII or binary or other type. The predefined string is **not limited** to containing ASCII or binary or other type. The predefined flag is **not limited** to whether it is a positive or negative number. The predefined flag is **not limited** to whether it is a decimal number or not.

performing a bitwise exclusive OR operation between an ASCII binary representation of at least a segment of the unknown string and an ASCII binary representation of at least a segment of the selected predefined string;

performing a bitwise operation between a predefined flag and a result of the exclusive OR operation; and

comparing the predefined predetermined flag and a result of the bitwise operation to produce an indication for the a case-insensitive string match, wherein the indication for the case-

Hence, the above steps are **not sufficient** to accomplish case-insensitive string match between two strings as claimed in claims 1 and 26.

Similarly, below steps are **not sufficient** to accomplish case-insensitive string match between two headers as claimed in claim 25.

performing a bitwise exclusive OR operation on binary representations of the client HTTP header[[s]] and the known HTTP header selected from the database;

performing a bitwise OR operation between a result of the exclusive OR operation and a predetermined flag; and

comparing the predetermined flag and a result of the bitwise OR operation to produce an indication for a case-insensitive string match between the client HTTP header and the selected

Similarly, below steps are **not sufficient** to accomplish case-insensitive string match between two strings as claimed in claim 24.

performing at least one bitwise exclusive OR operation between characters of {[a]} the selected predefined string and corresponding characters of the unknown string,

performing a bitwise OR operation between a results of the bitwise exclusive OR operation and a predetermined flag; and

comparing the predetermined flag and a result of the bitwise OR operation to produce a single bit output that indicates whether a case-insensitive match exists between the selected

For example, if predefined string is "If-modified-since" (HTTP header), unknown string is "if-MODIFIED-SINCE", **predetermined flag is "0"** than the case-insensitive string match would indicate incorrect result that "If-modified-since" and "if-MODIFIED-SINCE " are not same. If predefined string is "Content-length" (HTTP header), unknown string is "cONTENT-LENGTH", **predetermined flag is "1"** than the case-insensitive string match would indicate incorrect result that "Content-length" and "cONTENT-LENGTH" are not same. If predefined string is "If-modified-since" (HTTP header), unknown string is "if-MATCH", **predetermined flag is "0"** than the case-insensitive string match would indicate incorrect result that "If-modified-since" and "if-MATCH" are same. If predefined string is "Content-length" (HTTP header), unknown string is "cONTENT-LOCATION", **predetermined flag is "1"** than the case-insensitive string match would indicate incorrect result that "Content-length" and "cONTENT-LOCATION" are same. Note: If the **predefined flag is a character or a group of characters**, then a single bit output cannot be produced as claimed. Further, the predefined flag can be of any size, 4 byte flag, 8 bit flag, or any other size flag.

Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. **See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).** The First inquiry must be into exactly what the **claims** define. **See *In re Wilder*, 166 USPQ 545, 548 (CCPA 1970).**

**Further, the specification of this application at page 16 contains below:**

> While the present invention has been particularly shown and described with reference to the foregoing preferred embodiments, those skilled in the art will understand that many variations may be made therein without departing from the spirit and scope of the invention as defined in the following claims. The description of the invention should be understood to include all novel and nonobvious combinations of elements described herein, and claims may be presented in this or a later application to any novel and nonobvious combination of these elements. Where the claims recite "a" or "a first" element or the equivalent thereof, such claims should be understood to include incorporation of one or more such elements, neither requiring nor excluding two or more such elements.

The applicant's below statements at page 6 of the remarks 2/18/2010 are noted. It is also noted that these statements are **not regarding networking**. Note, the claims are **not limited** to that common case strings cannot be provided for the comparison; meaning when both the strings for comparison are already in a common case, the converting is not necessary at all. It is also noted that the claimed unknown string is not limited to a particular type of string.

> To further aid the Examiner understanding of the claimed invention, Applicant submits that there may be many algorithms for performing a case-insensitive comparison of a known string to an unknown string. Many algorithms may rely on first converting both strings to a common case, such as converting both to upper case or converting both to lower case. After converting the strings, the individual bytes of each string can then be compared to test for equality. These techniques, however, are costly in terms of memory cycles and CPU operations in that they require byte-wise conversion of each character in the string to a common case.[14] This conversion process may be performed by performing a comparison and then an addition or subtraction operation on each character. Applicant's claimed technique may avoid the disadvantages of these time-consuming approaches for converting the strings to a common case.

According, **several prior arts containing the case-insensitive comparison without converting the strings to a common case** are provided in this office action.

Further it is noted that: **XOR is the exclusive-OR operator is a well-known operation, which by definition can be expressed using AND and OR, for example: x X0R y == (x AND NOT y) OR (y AND NOT x).**

***Claim Rejections - 35 USC § 103***

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1-3, 5, 8, 14-17, 19-20, 22-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith et al. U. S. Publication 2005/0246716, Microsoft Corporation (Hereinafter Smith-Microsoft) in view of Larson et al., 6,381,616, Microsoft Corporation (Hereinafter Larson-Microsoft).

8. Referring to claim 1, Smith-Microsoft discloses a computer implemented method for comparing an unknown string to a predefined string (comparing of http headers, case insensitive string matching functions, paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045), the method comprising: storing, on a network device a database containing a plurality of predefined string (usage of HTTP

Authentication server, HTTP servers, HTTP database, HTTP Clients, HTTP browsers, etc., paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045), wherein the predefined strings stored within the database represent known headers for a network communication protocol (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045); receiving, with the network device, a network message in response to receiving the network message (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045), selecting one of the plurality of predefined strings stored within the database of the network device (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045); identifying a portion of the network message as an unknown string for comparison with the selected predefined string (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045); performing an operation between an ASCII binary representation of at least a segment of the unknown string and an ASCII binary representation of at least a segment of the selected predefined string (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045); produce an indication for a case-insensitive string match wherein the indication for the case-insensitive string match indicates whether all characters of the unknown string within the network message match all corresponding characters of the selected predefined string so as to match one of the known headers of the network communication protocol (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045, http strings are case insensitive and http strings comparison produce result); processing the network message based on the indication of a case

insensitive string match (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045, http strings are case insensitive and http strings comparison contain operations); and outputting a response from the network device based on the processed network message (usage of HTTP Authentication server, HTTP servers, HTTP database, HTTP Clients, HTTP browsers, etc., comparing of http headers, case insensitive string matching functions, paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045).

Smith-Microsoft does not disclose performing a bitwise exclusive OR operation and performing a bitwise operation between a predefined flag and a result of the exclusive OR operation; and comparing the predefined flag and a result of the bitwise operation. Larson-Microsoft discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., cols., 10-14. However, since Smith-Microsoft does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Smith-Microsoft is open for different techniques for string comparisons, and the ability to be adaptive to enables Smith-Microsoft's system/method/medium to be implemented in a wider variety of techniques including Larson-Microsoft's technique for string comparison.

Note: Regarding the applicant's usage of "wherein" and/or "whereby" and/or "adapted to" and/or "adapted for" in the claimed subject matter of the claims, the claim scope is not limited by claim language that suggests or makes optional but does not require steps to be performed, or by claim language that does not limit a claim to a particular structure. Please see

Minton v. Nat 'l Ass 'n of Securities Dealers, Inc., 336 F.3d 1373, 1381, 67 USPQ2d 1614, 1620 (Fed. Cir. 2003)), MPEP 2111.

9. Referring to claim 24, Smith-Microsoft discloses a method of case-insensitive string matching for use in a computer network (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045), the method comprising: storing, on a network device, a plurality of predefined strings, wherein the predefined strings represent known headers for a network communication protocol (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045); receiving with the network device, the network message selecting one of the plurality of predefined strings stored within the network device (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045); identifying a portion of the network message as an unknown string for comparison with the selected predefined string (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045); indicates whether a case-insensitive match exists between the selected predefined string and the unknown string (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045, http strings are case insensitive and http strings comparison contain operations); processing the network message based on the indication of the case-insensitive match (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045); and outputting a response from the network device (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045).

Smith-Microsoft does not disclose performing a bitwise exclusive OR operation and performing a bitwise OR operation between a results of the bitwise exclusive OR operation and a predetermined flag; and comparing the predetermined flag and a result of the bitwise OR operation to produce a single bit output. Larson-Microsoft discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., cols., 10-14. However, since Smith-Microsoft does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Smith-Microsoft is open for different techniques for string comparisons, and the ability to be adaptive to enables Smith-Microsoft's system/method/medium to be implemented in a wider variety of techniques including Larson-Microsoft's technique for string comparison.

10. Referring to claim 25, Smith-Microsoft discloses a computer networking device for improving data transfer via a computer network, the device comprising a processor configured to compare a client HTTP header with a known HTTP header by storing, on the networking device, a database containing a plurality of known HTTP headers(paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045); receiving with the networking device, a client HTTP header in response to receiving the client HTTP header, selecting one of the known HTTP headers stored within the database of the network device (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045); to produce an indication for a case-insensitive string match between the client HTTP header (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045) and the selected known HTTP header

predefined processing the network message based on the indication of the case-insensitive match and outputting a response from the network device based on the processed network message (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045, http strings are case insensitive and http strings comparison contain operations).

Smith-Microsoft does not disclose performing a bitwise exclusive OR operation and performing a bitwise OR operation between a result of the exclusive OR, operation and a predetermined flag; and comparing the predetermined flag and a result of the bitwise OR operation to produce an indication. Larson-Microsoft discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., cols., 10-14. However, since Smith-Microsoft does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Smith-Microsoft is open for different techniques for string comparisons, and the ability to be adaptive to enables Smith-Microsoft's system/method/medium to be implemented in a wider variety of techniques including Larson-Microsoft's technique for string comparison.

11. Referring to claim 26, Smith-Microsoft an article of manufacture comprising a storage medium having a plurality of machine-readable instructions, wherein when the instructions arc executed by a computing system, the instructions providing for: storing, on a network device, a database containing a plurality of predefined strings (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045), wherein the predefined strings stored, within the database represent known headers for a network

communication protocol receiving with the network device (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045), a network message in response to receiving the network message (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045), selecting one of the plurality of predefined strings, stored within the database of the network device (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045); identifying a portion of the network message as an unknown string for comparison with the selected predefined string (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045); performing an operation between an ASCII binary representation of at least a segment of the unknown string and an ASCII binary representation of at least a segment of the selected predefined string (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045) to produce an indication for a case-insensitive string match between the predefined string and the unknown string (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045, http strings are case insensitive and http strings comparison contain operations), wherein the indication for the case-insensitive match indicates whether all characters of the unknown string within the network message match all corresponding characters of the identified predefined string so as to match one of the known headers of the network communication protocol (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045); processing the network message based on the indication of the case-insensitive match and outputting a response from the network device based on the processed network message (paragraphs, 28768,

28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045).

Smith-Microsoft does not disclose performing a bitwise exclusive OR operation and performing a bitwise operation between a predefined flag and a result of the exclusive OR operation; and comparing the predefined flag and a result of the bitwise operation. Larson-Microsoft discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., cols., 10-14. However, since Smith-Microsoft does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Smith-Microsoft is open for different techniques for string comparisons, and the ability to be adaptive to enables Smith-Microsoft's system/method/medium to be implemented in a wider variety of techniques including Larson-Microsoft's technique for string comparison.

12. Referring to claim 2, Smith-Microsoft and Larson-Microsoft discloses the claimed limitations as rejected above. Smith-Microsoft also discloses identifying a segment of the selected predefined string and identifying a segment of the unknown string for comparison with the identified segment of the selected predefined string (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045).

13. Referring to claim 3, Smith-Microsoft and Larson-Microsoft discloses the claimed limitations as rejected above. Smith-Microsoft also discloses wherein the segment of the selected predefined string and the segment of the unknown string contain a same number of characters

(paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045).

14. Referring to claim 5, Smith-Microsoft and Larson-Microsoft discloses the claimed limitations as rejected above. Larson-Microsoft also discloses wherein identifying a case-insensitive string match includes identifying a case-insensitive segment match based on the exclusive OR operation, e.g., cols., 10-14.

15. Referring to claim 8, Smith-Microsoft and Larson-Microsoft discloses the claimed limitations as rejected above. Larson-Microsoft also discloses identifying a subsequent segment of the selected predefined string and a subsequent segment oft he unknown string for comparison, e.g., cols., 10-14.

16. Referring to claim 14, Smith-Microsoft and Larson-Microsoft discloses the claimed limitations as rejected above. Smith-Microsoft also discloses wherein the segments of the unknown string and the segment of the selected predefined string each include one character (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045).

17. Referring to claim 15, Smith-Microsoft and Larson-Microsoft discloses the claimed limitations as rejected above. Smith-Microsoft also discloses wherein the segments of the unknown string and the segment of the selected predefined string each include four characters

(paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045).

18. Referring to claim 16, Smith-Microsoft and Larson-Microsoft discloses the claimed limitations as rejected above. Smith-Microsoft also discloses wherein the unknown string includes an HTTP header field (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045).

19. Referring to claim 17, Smith-Microsoft and Larson-Microsoft discloses the claimed limitations as rejected above. Smith-Microsoft also discloses wherein the selected predefined string is from a table of predetermined HTTP header fields (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045).

20. Referring to claim 19, Smith-Microsoft and Larson-Microsoft discloses the claimed limitations as rejected above. Smith-Microsoft also discloses identifying the length of the strings (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045).

21. Referring to claim 20, Smith-Microsoft and Larson-Microsoft discloses the claimed limitations as rejected above. Smith-Microsoft also discloses wherein the length of each of the strings are equal (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045).

22. Referring to claim 22, Smith-Microsoft and Larson-Microsoft discloses the claimed limitations as rejected above. Smith-Microsoft also discloses determining if characters of the strings are within a predefined ASCII range (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045).

23. Referring to claim 23, Smith-Microsoft and Larson-Microsoft discloses the claimed limitations as rejected above. Smith-Microsoft also discloses wherein characters not within the predefined ASCII range caused the method to yield a negative string match (paragraphs, 28768, 28759, 19857, 19860-19864, 19874-19899, 68, 77, 14441, 14435, 20109-20115, 22, 30552, 29045).

24. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Smith-Microsoft in view of Larson-Microsoft and Thinkage GCOS8 SS C Reference Manual, pages 1-71, 1996 (Hereinafter Thinkage).

25. Referring to claim 4, Smith-Microsoft and Larson-Microsoft discloses the claimed limitations as rejected above. Smith-Microsoft and Larson-Microsoft does not specifically mention about shifting when less than four characters, which is disclosed by Thinkage, e.g., section, 2.7, page 6, section, 4.7, page 34, section, 4.12, page 36.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the well-known concept of shifting when less than four characters with the teachings of Smith-Microsoft, Larson-Microsoft and Thinkage in order to facilitate shifting when

less than four characters exist because the shifting would enhance comparison of strings content. The compared information would be used for determining case insensitive match.

26. Claims 7, 11-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith-Microsoft in view of Larson-Microsoft and Abgrall et al., 2003/0037237 (Hereinafter Abgrall).

27. Referring to claims 7, 11-13, Smith-Microsoft and Larson-Microsoft discloses the claimed limitations as rejected above. Smith-Microsoft and Larson-Microsoft does not specifically mention about predetermined value 0x20202020 / 0x20 / 0, which is disclosed by Abgrall, 0x20 for each byte, e.g., paragraphs 323 and 324. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the well-known concept of providing predetermined value 0x20202020 / 0x20 / 0 with the teachings of Smith-Microsoft, Larson-Microsoft and Abgrall in order to facilitate usage of 0x20202020 / 0x20 / 0 because it would enhance representing four blank characters / single blank character / null value for comparison of strings content. The compared information would be used for determining case insensitive match.

28. Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Smith-Microsoft in view of Larson-Microsoft and Kontio et al., 2005/0004875 (Hereinafter Kontio).

29. Referring to claim 18, Smith-Microsoft and Larson-Microsoft discloses the claimed limitations as rejected above. Smith-Microsoft and Larson-Microsoft does not specifically mention about providing further bitwise operation, which is disclosed by Kontio, e.g., paragraphs 54 and 55. It would have been obvious to one of ordinary skill in the art at the time the invention

was made to include the well-known concept of providing further bitwise operation with the teachings of Smith-Microsoft, Larson-Microsoft and Kontio in order to facilitate usage of further bitwise operation because it would enhance comparison of strings content performed with the bitwise operation. The compared information would be used for determining case insensitive match.

30. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Smith-Microsoft in view of Larson-Microsoft and Slater et al., 6,654,796, Cisco (Hereinafter Slater).

31. Referring to claim 21, Smith-Microsoft and Larson-Microsoft discloses the claimed limitations as rejected above. Smith-Microsoft and Larson-Microsoft does not specifically mention about the network being WAN, which is disclosed by Slater, e.g., col., 1, lines 55 – 67, col., 9, lines 42 – 65. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the well-known concept of providing the WAN with the teachings of Smith-Microsoft, Larson-Microsoft and Slater in order to facilitate usage of the WAN because it would enhance receiving strings for comparison from different networks. The compared information of the strings from different networks would be used for determining case insensitive match.

32. Claims 1-3, 5, 8, 14-17, 19-20, 22-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Knouse et al. U. S. Publication 7,249,369, Oracle International Corporation (Hereinafter Knouse-Oracle) in view of James et al., 6,523,108 (Hereinafter James).

33. Referring to claim 1, Knouse-Oracle discloses a computer implemented method for comparing an unknown string to a predefined string (comparing of segment or all portions of HTTP Post data with policy matching data, usage of integrated circuits, gate arrays, hardware circuit for comparing of http headers, cols., 3, 26, 27, 49), the method comprising: storing, on a network device a database containing a plurality of predefined string (usage of HTTP Authentication server, HTTP servers, HTTP database, HTTP Clients, HTTP browsers, etc., cols., 3, 26, 27, 49), wherein the predefined strings stored within the database represent known headers for a network communication protocol (cols., 3, 26, 27, 49); receiving, with the network device, a network message in response to receiving the network message (cols., 3, 26, 27, 49), selecting one of the plurality of predefined strings stored within the database of the network device (cols., 3, 26, 27, 49); identifying a portion of the network message as an unknown string for comparison with the selected predefined string (cols., 3, 26, 27, 49); performing an operation between an ASCII binary representation of at least a segment of the unknown string and an ASCII binary representation of at least a segment of the selected predefined string (cols., 3, 26, 27, 49); produce an indication for a case-insensitive string match wherein the indication for the case-insensitive string match indicates whether all characters of the unknown string within the network message match all corresponding characters of the selected predefined string so as to match one of the known headers of the network communication protocol (cols., 3, 26, 27, 49, http strings are case insensitive and http strings comparison produce result); processing the network message based on the indication of a case insensitive string match (cols., 3, 26, 27, 49, http strings are case insensitive and http strings comparison contain operations); and outputting a response from the network device based on the processed network message (usage of HTTP

Authentication server, HTTP servers, HTTP database, HTTP Clients, HTTP browsers, etc., comparing of segment or all portions of HTTP Post data with policy matching data, usage of integrated circuits, gate arrays, hardware circuit for comparing of http headers, cols., 3, 26, 27, 49).

Knouse-Oracle does not disclose performing a bitwise exclusive OR operation and performing a bitwise operation between a predefined flag and a result of the exclusive OR operation; and comparing the predefined flag and a result of the bitwise operation. James discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., cols., 3, 4. However, since Knouse-Oracle does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Knouse-Oracle is open for different techniques for string comparisons, and the ability to be adaptive to enables Knouse-Oracle's system/method/medium to be implemented in a wider variety of techniques including James's bitwise technique for string comparison.

34. Referring to claim 24, Knouse-Oracle discloses a method of case-insensitive string matching for use in a computer network (cols., 3, 26, 27, 49), the method comprising: storing, on a network device, a plurality of predefined strings, wherein the predefined strings represent known headers for a network communication protocol (cols., 3, 26, 27, 49); receiving with the network device, the network message selecting one of the plurality of predefined strings stored within the network device (cols., 3, 26, 27, 49); identifying a portion of the network message as an unknown string for comparison with the selected predefined string (cols., 3, 26, 27, 49); indicates whether a case-insensitive match exists between the selected predefined string and the

unknown string (cols., 3, 26, 27, 49, http strings are case insensitive and http strings comparison contain operations); processing the network message based on the indication of the case-insensitive match (cols., 3, 26, 27, 49); and outputting a response from the network device (cols., 3, 26, 27, 49).

Knouse-Oracle does not disclose performing a bitwise exclusive OR operation and performing a bitwise OR operation between a results of the bitwise exclusive OR operation and a predetermined flag; and comparing the predetermined flag and a result of the bitwise OR operation to produce a single bit output. James discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., cols., 3, 4. However, since Knouse-Oracle does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Knouse-Oracle is open for different techniques for string comparisons, and the ability to be adaptive to enables Knouse-Oracle's system/method/medium to be implemented in a wider variety of techniques including James's bitwise technique for string comparison.

35. Referring to claim 25, Knouse-Oracle discloses a computer networking device for improving data transfer via a computer network, the device comprising a processor configured to compare a client HTTP header with a known HTTP header by storing, on the networking device, a database containing a plurality of known HTTP headers(cols., 3, 26, 27, 49); receiving with the networking device, a client HTTP header in response to receiving the client HTTP header, selecting one of the known HTTP headers stored within the database of the network device (cols., 3, 26, 27, 49); to produce an indication for a case-insensitive string match between the

client HTTP header (cols., 3, 26, 27, 49) and the selected known HTTP header predefined processing the network message based on the indication of the case-insensitive match and outputting a response from the network device based on the processed network message (cols., 3, 26, 27, 49, http strings are case insensitive and http strings comparison contain operations).

Knouse-Oracle does not disclose performing a bitwise exclusive OR operation and performing a bitwise OR operation between a result of the exclusive OR, operation and a predetermined flag; and comparing the predetermined flag and a result of the bitwise OR operation to produce an indication. James discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., cols., 3, 4. However, since Knouse-Oracle does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Knouse-Oracle is open for different techniques for string comparisons, and the ability to be adaptive to enables Knouse-Oracle's system/method/medium to be implemented in a wider variety of techniques including James's bitwise technique for string comparison.

36. Referring to claim 26, Knouse-Oracle an article of manufacture comprising a storage medium having a plurality of machine-readable instructions, wherein when the instructions arc executed by a computing system, the instructions providing for: storing, on a network device, a database containing a plurality of predefined strings (cols., 3, 26, 27, 49), wherein the predefined strings stored, within the database represent known headers for a network communication protocol receiving with the network device (cols., 3, 26, 27, 49), a network message in response to receiving the network message (cols., 3, 26, 27, 49), selecting one of the plurality of

predefined strings, stored within the database of the network device (cols., 3, 26, 27, 49); identifying a portion of the network message as an unknown string for comparison with the selected predefined string (cols., 3, 26, 27, 49); performing an operation between an ASCII binary representation of at least a segment of the unknown string and an ASCII binary representation of at least a segment of the selected predefined string (cols., 3, 26, 27, 49); to produce an indication for a case-insensitive string match between the predefined string and the unknown string (cols., 3, 26, 27, 49, http strings are case insensitive and http strings comparison contain operations), wherein the indication for the case-insensitive match indicates whether all characters of the unknown string within the network message match all corresponding characters of the identified predefined string so as to match one of the known headers of the network communication protocol (cols., 3, 26, 27, 49); processing the network message based on the indication of the case-insensitive match and outputting a response from the network device based on the processed network message (cols., 3, 26, 27, 49).

Knouse-Oracle does not disclose performing a bitwise exclusive OR operation and performing a bitwise operation between a predefined flag and a result of the exclusive OR operation; and comparing the predefined flag and a result of the bitwise operation. James discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., cols., 3, 4. However, since Knouse-Oracle does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Knouse-Oracle is open for different techniques for string comparisons, and the ability to be adaptive to enables Knouse-Oracle's system/method/medium to be implemented in a wider variety of techniques including James's bitwise technique for string comparison.

37. Referring to claim 2, Knouse-Oracle and James discloses the claimed limitations as rejected above. Knouse-Oracle also discloses identifying a segment of the selected predefined string and identifying a segment of the unknown string for comparison with the identified segment of the selected predefined string (cols., 3, 26, 27, 49).

38. Referring to claim 3, Knouse-Oracle and James discloses the claimed limitations as rejected above. Knouse-Oracle also discloses wherein the segment of the selected predefined string and the segment of the unknown string contain a same number of characters (cols., 3, 26, 27, 49).

39. Referring to claim 5, Knouse-Oracle and James discloses the claimed limitations as rejected above. James also discloses wherein identifying a case-insensitive string match includes identifying a case-insensitive segment match based on the exclusive OR operation, e.g., cols., 10-14.

40. Referring to claim 8, Knouse-Oracle and James discloses the claimed limitations as rejected above. James also discloses identifying a subsequent segment of the selected predefined string and a subsequent segment oft he unknown string for comparison, e.g., cols., 10-14.

41. Referring to claim 14, Knouse-Oracle and James discloses the claimed limitations as rejected above. Knouse-Oracle also discloses wherein the segments of the unknown string and the segment of the selected predefined string each include one character (cols., 3, 26, 27, 49).

42. Referring to claim 15, Knouse-Oracle and James discloses the claimed limitations as rejected above. Knouse-Oracle also discloses wherein the segments of the unknown string and the segment of the selected predefined string each include four characters (cols., 3, 26, 27, 49).

43. Referring to claim 16, Knouse-Oracle and James discloses the claimed limitations as rejected above. Knouse-Oracle also discloses wherein the unknown string includes an HTTP header field (cols., 3, 26, 27, 49).

44. Referring to claim 17, Knouse-Oracle and James discloses the claimed limitations as rejected above. Knouse-Oracle also discloses wherein the selected predefined string is from a table of predetermined HTTP header fields (cols., 3, 26, 27, 49).

45. Referring to claim 19, Knouse-Oracle and James discloses the claimed limitations as rejected above. Knouse-Oracle also discloses identifying the length of the strings (cols., 3, 26, 27, 49).

46. Referring to claim 20, Knouse-Oracle and James discloses the claimed limitations as rejected above. Knouse-Oracle also discloses wherein the length of each of the strings are equal (cols., 3, 26, 27, 49).

47. Referring to claim 22, Knouse-Oracle and James discloses the claimed limitations as rejected above. Knouse-Oracle also discloses determining if characters of the strings are within a predefined ASCII range (cols., 3, 26, 27, 49).

48. Referring to claim 23, Knouse-Oracle and James discloses the claimed limitations as rejected above. Knouse-Oracle also discloses wherein characters not within the predefined ASCII range caused the method to yield a negative string match (cols., 3, 26, 27, 49).

49. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Knouse-Oracle in view of James and Thinkage GCOS8 SS C Reference Manual, pages 1-71, 1996 (Hereinafter Thinkage).

50. Referring to claim 4, Knouse-Oracle and James discloses the claimed limitations as rejected above. Knouse-Oracle and James does not specifically mention about shifting when less than four characters, which is disclosed by Thinkage, e.g., section, 2.7, page 6, section, 4.7, page 34, section, 4.12, page 36.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the well-known concept of shifting when less than four characters with the teachings of Knouse-Oracle, James and Thinkage in order to facilitate shifting when less than

four characters exist because the shifting would enhance comparison of strings content. The compared information would be used for determining case insensitive match.

51. Claims 7, 11-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Knouse-Oracle in view of James and Abgrall et al., 2003/0037237 (Hereinafter Abgrall).

52. Referring to claims 7, 11-13, Knouse-Oracle and James discloses the claimed limitations as rejected above. Knouse-Oracle and James does not specifically mention about predetermined value 0x20202020 / 0x20 / 0, which is disclosed by Abgrall, 0x20 for each byte, e.g., paragraphs 323 and 324. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the well-known concept of providing predetermined value 0x20202020 / 0x20 / 0 with the teachings of Knouse-Oracle, James and Abgrall in order to facilitate usage of 0x20202020 / 0x20 / 0 because it would enhance representing four blank characters / single blank character / null value for comparison of strings content. The compared information would be used for determining case insensitive match.

53. Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Knouse-Oracle in view of James and Kontio et al., 2005/0004875 (Hereinafter Kontio).

54. Referring to claim 18, Knouse-Oracle and James discloses the claimed limitations as rejected above. Knouse-Oracle and James does not specifically mention about providing further bitwise operation, which is disclosed by Kontio, e.g., paragraphs 54 and 55. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the well-known concept of providing further bitwise operation with the teachings of Knouse-Oracle,

James and Kontio in order to facilitate usage of further bitwise operation because it would enhance comparison of strings content performed with the bitwise operation. The compared information would be used for determining case insensitive match.

55. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Knouse-Oracle in view of James and Slater et al., 6,654,796, Cisco (Hereinafter Slater).

56. Referring to claim 21, Knouse-Oracle and James discloses the claimed limitations as rejected above. Knouse-Oracle and James does not specifically mention about the network being WAN, which is disclosed by Slater, e.g., col., 1, lines 55 – 67, col., 9, lines 42 – 65. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the well-known concept of providing the WAN with the teachings of Knouse-Oracle, James and Slater in order to facilitate usage of the WAN because it would enhance receiving strings for comparison from different networks. The compared information of the strings from different networks would be used for determining case insensitive match.

57. Claims 1-3, 5, 8, 14-17, 19-20, 22-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Reiche et al. 6,092,196, Nortel Networks Limited (Hereinafter Reiche-Nortel-Networks) in view of Geist 5,329,598 (Hereinafter Geist).

58. Referring to claim 1, Reiche-Nortel-Networks discloses a computer implemented method for comparing an unknown string to a predefined string (col., 5, line 1 – col., 6, line 64), the method comprising: storing, on a network device a database containing a plurality of predefined

string (usage of HTTP Authentication server, HTTP servers, HTTP database, HTTP Clients, HTTP browsers, etc., figure 1, col., 5, line 1 – col., 6, line 64), wherein the predefined strings stored within the database represent known headers for a network communication protocol (col., 5, line 1 – col., 6, line 64); receiving, with the network device, a network message in response to receiving the network message (figure 1, col., 5, line 1 – col., 6, line 64), selecting one of the plurality of predefined strings stored within the database of the network device (col., 5, line 1 – col., 6, line 64); identifying a portion of the network message as an unknown string for comparison with the selected predefined string (figure 1, col., 5, line 1 – col., 6, line 64); performing an operation between an ASCII binary representation of at least a segment of the unknown string and an ASCII binary representation of at least a segment of the selected predefined string (figure 1, col., 5, line 1 – col., 6, line 64); produce an indication for a case-insensitive string match wherein the indication for the case-insensitive string match indicates whether all characters of the unknown string within the network message match all corresponding characters of the selected predefined string so as to match one of the known headers of the network communication protocol (col., 5, line 1 – col., 6, line 64, http strings are case insensitive and http strings comparison produce result); processing the network message based on the indication of a case insensitive string match (col., 5, line 1 – col., 6, line 64, http strings are case insensitive and http strings comparison contain operations); and outputting a response from the network device based on the processed network message (usage of HTTP Authentication server, HTTP servers, HTTP database, HTTP Clients, HTTP browsers, etc., figure 1, col., 5, line 1 – col., 6, line 64).

Reiche-Nortel-Networks does not disclose performing a bitwise exclusive OR operation and performing a bitwise operation between a predefined flag and a result of the exclusive OR operation; and comparing the predefined flag and a result of the bitwise operation. Geist discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., usage of XOR, etc., figures 9A-9D and its description. However, since Reiche-Nortel-Networks does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Reiche-Nortel-Networks is open for different techniques for string comparisons, and the ability to be adaptive to enables Reiche-Nortel-Networks's system/method/medium to be implemented in a wider variety of techniques including Geist's technique for string comparison.

59. Referring to claim 24, Reiche-Nortel-Networks discloses a method of case-insensitive string matching for use in a computer network (col., 5, line 1 – col., 6, line 64), the method comprising: storing, on a network device, a plurality of predefined strings, wherein the predefined strings represent known headers for a network communication protocol (col., 5, line 1 – col., 6, line 64); receiving with the network device, the network message selecting one of the plurality of predefined strings stored within the network device (col., 5, line 1 – col., 6, line 64); identifying a portion of the network message as an unknown string for comparison with the selected predefined string (col., 5, line 1 – col., 6, line 64); indicates whether a case-insensitive match exists between the selected predefined string and the unknown string (col., 5, line 1 – col., 6, line 64, http strings are case insensitive and http strings comparison contain operations); processing the network message based on the indication of the case-insensitive match (col., 5,

line 1 – col., 6, line 64); and outputting a response from the network device (col., 5, line 1 – col., 6, line 64).

Reiche-Nortel-Networks does not disclose performing a bitwise exclusive OR operation and performing a bitwise OR operation between a results of the bitwise exclusive OR operation and a predetermined flag; and comparing the predetermined flag and a result of the bitwise OR operation to produce a single bit output. Geist discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., usage of XOR, etc., figures 9A-9D and its description. However, since Reiche-Nortel-Networks does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Reiche-Nortel-Networks is open for different techniques for string comparisons, and the ability to be adaptive to enables Reiche-Nortel-Networks's system/method/medium to be implemented in a wider variety of techniques including Geist's technique for string comparison.

60. Referring to claim 25, Reiche-Nortel-Networks discloses a computer networking device for improving data transfer via a computer network, the device comprising a processor configured to compare a client HTTP header with a known HTTP header by storing, on the networking device, a database containing a plurality of known HTTP headers(col., 5, line 1 – col., 6, line 64); receiving with the networking device, a client HTTP header in response to receiving the client HTTP header, selecting one of the known HTTP headers stored within the database of the network device (col., 5, line 1 – col., 6, line 64); to produce an indication for a case-insensitive string match between the client HTTP header (col., 5, line 1 – col., 6, line 64)

and the selected known HTTP header predefined processing the network message based on the indication of the case-insensitive match and outputting a response from the network device based on the processed network message (col., 5, line 1 – col., 6, line 64, http strings are case insensitive and http strings comparison contain operations).

Reiche-Nortel-Networks does not disclose performing a bitwise exclusive OR operation and performing a bitwise OR operation between a result of the exclusive OR, operation and a predetermined flag; and comparing the predetermined flag and a result of the bitwise OR operation. Geist discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., usage of XOR, etc., figures 9A-9D and its description. However, since Reiche-Nortel-Networks does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Reiche-Nortel-Networks is open for different techniques for string comparisons, and the ability to be adaptive to enables Reiche-Nortel-Networks's system/method/medium to be implemented in a wider variety of techniques including Geist's technique for string comparison.

61. Referring to claim 26, Reiche-Nortel-Networks an article of manufacture comprising a storage medium having a plurality of machine-readable instructions, wherein when the instructions arc executed by a computing system, the instructions providing for: storing, on a network device, a database containing a plurality of predefined strings (col., 5, line 1 – col., 6, line 64), wherein the predefined strings stored, within the database represent known headers for a network communication protocol receiving with the network device (col., 5, line 1 – col., 6, line 64), a network message in response to receiving the network message (col., 5, line 1 – col., 6,

line 64), selecting one of the plurality of predefined strings, stored within the database of the network device (col., 5, line 1 – col., 6, line 64); identifying a portion of the network message as an unknown string for comparison with the selected predefined string (col., 5, line 1 – col., 6, line 64); performing an operation between an ASCII binary representation of at least a segment of the unknown string and an ASCII binary representation of at least a segment of the selected predefined string (figure 1, col., 5, line 1 – col., 6, line 64); to produce an indication for a case-insensitive string match between the predefined string and the unknown string (col., 5, line 1 – col., 6, line 64, http strings are case insensitive and http strings comparison contain operations), wherein the indication for the case-insensitive match indicates whether all characters of the unknown string within the network message match all corresponding characters of the identified predefined string so as to match one of the known headers of the network communication protocol (col., 5, line 1 – col., 6, line 64); processing the network message based on the indication of the case-insensitive match and outputting a response from the network device based on the processed network message (col., 5, line 1 – col., 6, line 64).

Reiche-Nortel-Networks does not disclose performing a bitwise exclusive OR operation and performing a bitwise operation between a predefined flag and a result of the exclusive OR operation; and comparing the predefined flag and a result of the bitwise operation. Geist discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., usage of XOR, etc., figures 9A-9D and its description. However, since Reiche-Nortel-Networks does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Reiche-Nortel-Networks is open for different techniques for string comparisons, and the ability to be adaptive to enables

Reiche-Nortel-Networks's system/method/medium to be implemented in a wider variety of techniques including Geist's technique for string comparison.

62. Claims 1-3, 5, 8, 14-17, 19-20, 22-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Aviani, Jr. et al. 6,532,493, Cisco Technology (Hereinafter Aviani-Cisco-Technology) in view of Khieu, 5,381,127 (Hereinafter Khieu).

63. Referring to claim 1, Aviani-Cisco-Technology discloses a computer implemented method for comparing an unknown string to a predefined string (col., 6, line 20 - col., 8, line 26), the method comprising: storing, on a network device a database containing a plurality of predefined string (col., 6, line 20 - col., 8, line 26), wherein the predefined strings stored within the database represent known headers for a network communication protocol (col., 6, line 20 - col., 8, line 26); receiving, with the network device, a network message in response to receiving the network message (col., 6, line 20 - col., 8, line 26), selecting one of the plurality of predefined strings stored within the database of the network device (col., 6, line 20 - col., 8, line 26); identifying a portion of the network message as an unknown string for comparison with the selected predefined string ( col., 6, line 20 - col., 8, line 26); performing an operation between an ASCII binary representation of at least a segment of the unknown string and an ASCII binary representation of at least a segment of the selected predefined string ( col., 6, line 20 - col., 8, line 26); produce an indication for a case-insensitive string match wherein the indication for the case-insensitive string match indicates whether all characters of the unknown string within the network message match all corresponding characters of the selected predefined string so as to

match one of the known headers of the network communication protocol (col., 6, line 20 - col., 8, line 26, http strings are case insensitive and http strings comparison contain operations); processing the network message based on the indication of a case insensitive string match (col., 6, line 20 - col., 8, line 26); and outputting a response from the network device based on the processed network message (col., 6, line 20 - col., 8, line 26).

Aviani-Cisco-Technology does not disclose performing a bitwise exclusive OR operation and performing a bitwise operation between a predefined flag and a result of the exclusive OR operation; and comparing the predefined flag and a result of the bitwise operation. Khieu discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., usage of XOR, etc., figure 4 and its description. However, since Aviani-Cisco-Technology does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Aviani-Cisco-Technology is open for different techniques for string comparisons, and the ability to be adaptive to enables Aviani-Cisco-Technology's system/method/medium to be implemented in a wider variety of techniques including Khieu's technique for string comparison.

64. Referring to claim 24, Aviani-Cisco-Technology discloses a method of case-insensitive string matching for use in a computer network (col., 6, line 20 - col., 8, line 26), the method comprising: storing, on a network device, a plurality of predefined strings, wherein the predefined strings represent known headers for a network communication protocol (col., 6, line 20 - col., 8, line 26); receiving with the network device, the network message selecting one of the plurality of predefined strings stored within the network device (col., 6, line 20 - col., 8, line 26);

identifying a portion of the network message as an unknown string for comparison with the selected predefined string (col., 6, line 20 - col., 8, line 26); to produce a single bit output that indicates whether a case-insensitive match exists between the selected predefined string and the unknown string (col., 6, line 20 - col., 8, line 26, http strings are case insensitive and http strings comparison contain operations); processing the network message based on the indication of the case-insensitive match (col., 6, line 20 - col., 8, line 26); and outputting a response from the network device (col., 6, line 20 - col., 8, line 26).

Aviani-Cisco-Technology does not disclose performing a bitwise exclusive OR operation and performing a bitwise OR operation between a results of the bitwise exclusive OR operation and a predetermined flag and comparing the predetermined flag and a result of the bitwise OR operation. Khieu discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., usage of XOR, etc., figure 4 and its description. However, since Aviani-Cisco-Technology does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Aviani-Cisco-Technology is open for different techniques for string comparisons, and the ability to be adaptive to enables Aviani-Cisco-Technology's system/method/medium to be implemented in a wider variety of techniques including Khieu's technique for string comparison.

65. Referring to claim 25, Aviani-Cisco-Technology discloses a computer networking device for improving data transfer via a computer network, the device comprising a processor configured to compare a client HTTP header with a known HTTP header by storing, on the networking device, a database containing a plurality of known HTTP headers(col., 6, line 20 -

col., 8, line 26); receiving with the networking device, a client HTTP header in response to receiving the client HTTP header, selecting one of the known HTTP headers stored within the database of the network device (col., 6, line 20 - col., 8, line 26); to produce an indication for a case-insensitive string match between the client HTTP header (col., 6, line 20 - col., 8, line 26, http strings are case insensitive and http strings comparison contain operations) and the selected known HTTP header predefined processing the network message based on the indication of the case-insensitive match and outputting a response from the network device based on the processed network message (col., 6, line 20 - col., 8, line 26).

Aviani-Cisco-Technology does not disclose performing a bitwise exclusive OR operation and performing a bitwise OR operation between a result of the exclusive OR, operation and a predetermined flag; and comparing the predetermined flag and a result of the bitwise OR operation. Khieu discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., usage of XOR, etc., figure 4 and its description. However, since Aviani-Cisco-Technology does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Aviani-Cisco-Technology is open for different techniques for string comparisons, and the ability to be adaptive to enables Aviani-Cisco-Technology's system/method/medium to be implemented in a wider variety of techniques including Khieu's technique for string comparison.

66. Referring to claim 26, Aviani-Cisco-Technology an article of manufacture comprising a storage medium having a plurality of machine-readable instructions, wherein when the instructions arc executed by a computing system, the instructions providing for: storing, on a

network device, a database containing a plurality of predefined strings (col., 6, line 20 - col., 8, line 26), wherein the predefined strings stored, within the database represent known headers for a network communication protocol receiving with the network device (col., 6, line 20 - col., 8, line 26), a network message in response to receiving the network message (col., 6, line 20 - col., 8, line 26), selecting one of the plurality of predefined strings, stored within the database of the network device (col., 6, line 20 - col., 8, line 26); identifying a portion of the network message as an unknown string for comparison with the selected predefined string (col., 6, line 20 - col., 8, line 26); performing an operation between an ASCII binary representation of at least a segment of the unknown string and an ASCII binary representation of at least a segment of the selected predefined string ( col., 6, line 20 - col., 8, line 26); to produce an indication for a case-insensitive string match between the predefined string and the unknown string (col., 6, line 20 - col., 8, line 26, http strings are case insensitive and http strings comparison contain operations), wherein the indication for the case-insensitive match indicates whether all characters of the unknown string within the network message match all corresponding characters of the identified predefined string so as to match one of the known headers of the network communication protocol (col., 6, line 20 - col., 8, line 26); processing the network message based on the indication of the case-insensitive match and outputting a response from the network device based on the processed network message (col., 6, line 20 - col., 8, line 26).

Aviani-Cisco-Technology does not disclose performing a bitwise exclusive OR operation and performing a bitwise operation between a predefined flag and a result of the exclusive OR operation; and comparing the predefined flag and a result of the bitwise operation. Khieu discloses these limitations and well known techniques of comparing case-insensitive strings

without converting the strings to a common case, e.g., usage of XOR, etc., figure 4 and its description. However, since Aviani-Cisco-Technology does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Aviani-Cisco-Technology is open for different techniques for string comparisons, and the ability to be adaptive to enables Aviani-Cisco-Technology's system/method/medium to be implemented in a wider variety of techniques including Khieu's technique for string comparison.

67. Referring to claim 2, Aviani-Cisco-Technology and Khieu discloses the claimed limitations as rejected above. Aviani-Cisco-Technology also discloses identifying a segment of the selected predefined string and identifying a segment of the unknown string for comparison with the identified segment of the selected predefined string (col., 6, line 20 - col., 8, line 26).

68. Referring to claim 3, Aviani-Cisco-Technology and Khieu discloses the claimed limitations as rejected above. Aviani-Cisco-Technology also discloses wherein the segment of the selected predefined string and the segment of the unknown string contain a same number of characters (col., 6, line 20 - col., 8, line 26).

69. Referring to claim 5, Aviani-Cisco-Technology and Khieu discloses the claimed limitations as rejected above. Khieu also discloses wherein identifying a case-insensitive string match includes identifying a case-insensitive segment match based on the exclusive OR operation, e.g., cols., 10-14.

70. Referring to claim 8, Aviani-Cisco-Technology and Khieu discloses the claimed limitations as rejected above. Khieu also discloses identifying a subsequent segment of the selected predefined string and a subsequent segment oft he unknown string for comparison, e.g., cols., 10-14.

71. Referring to claim 14, Aviani-Cisco-Technology and Khieu discloses the claimed limitations as rejected above. Aviani-Cisco-Technology also discloses wherein the segments of the unknown string and the segment of the selected predefined string each include one character (col., 6, line 20 - col., 8, line 26).

72. Referring to claim 15, Aviani-Cisco-Technology and Khieu discloses the claimed limitations as rejected above. Aviani-Cisco-Technology also discloses wherein the segments of the unknown string and the segment of the selected predefined string each include four characters (col., 6, line 20 - col., 8, line 26).

73. Referring to claim 16, Aviani-Cisco-Technology and Khieu discloses the claimed limitations as rejected above. Aviani-Cisco-Technology also discloses wherein the unknown string includes an HTTP header field (col., 6, line 20 - col., 8, line 26).

74. Referring to claim 17, Aviani-Cisco-Technology and Khieu discloses the claimed limitations as rejected above. Aviani-Cisco-Technology also discloses wherein the selected

predefined string is from a table of predetermined HTTP header fields (col., 6, line 20 - col., 8, line 26).

75. Referring to claim 19, Aviani-Cisco-Technology and Khieu discloses the claimed limitations as rejected above. Aviani-Cisco-Technology also discloses identifying the length of the strings (col., 6, line 20 - col., 8, line 26).

76. Referring to claim 20, Aviani-Cisco-Technology and Khieu discloses the claimed limitations as rejected above. Aviani-Cisco-Technology also discloses wherein the length of each of the strings are equal (col., 6, line 20 - col., 8, line 26).

77. Referring to claim 22, Aviani-Cisco-Technology and Khieu discloses the claimed limitations as rejected above. Aviani-Cisco-Technology also discloses determining if characters of the strings are within a predefined ASCII range (col., 6, line 20 - col., 8, line 26).

78. Referring to claim 23, Aviani-Cisco-Technology and Khieu discloses the claimed limitations as rejected above. Aviani-Cisco-Technology also discloses wherein characters not within the predefined ASCII range caused the method to yield a negative string match (col., 6, line 20 - col., 8, line 26).

79. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Aviani-Cisco-Technology in view of Khieu and Thinkage GCOS8 SS C Reference Manual, pages 1-71, 1996 (Hereinafter Thinkage).

80. Referring to claim 4, Aviani-Cisco-Technology and Khieu discloses the claimed limitations as rejected above. Aviani-Cisco-Technology and Khieu does not specifically mention about shifting when less than four characters, which is disclosed by Thinkage, e.g., section, 2.7, page 6, section, 4.7, page 34, section, 4.12, page 36.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the well-known concept of shifting when less than four characters with the teachings of Aviani-Cisco-Technology, Khieu and Thinkage in order to facilitate shifting when less than four characters exist because the shifting would enhance comparison of strings content. The compared information would be used for determining case insensitive match.

81. Claims 7, 11-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Aviani-Cisco-Technology in view of Khieu and Abgrall et al., 2003/0037237 (Hereinafter Abgrall).

82. Referring to claims 7, 11-13, Aviani-Cisco-Technology and Khieu discloses the claimed limitations as rejected above. Aviani-Cisco-Technology and Khieu does not specifically mention about predetermined value 0x20202020 / 0x20 / 0, which is disclosed by Abgrall, 0x20 for each byte, e.g., paragraphs 323 and 324. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the well-known concept of providing predetermined value 0x20202020 / 0x20 / 0 with the teachings of Aviani-Cisco-Technology, Khieu and Abgrall in order to facilitate usage of 0x20202020 / 0x20 / 0 because it would

enhance representing four blank characters / single blank character / null value for comparison of strings content. The compared information would be used for determining case insensitive match.

83. Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Aviani-Cisco-Technology in view of Khieu and Kontio et al., 2005/0004875 (Hereinafter Kontio).

84. Referring to claim 18, Aviani-Cisco-Technology and Khieu discloses the claimed limitations as rejected above. Aviani-Cisco-Technology and Khieu does not specifically mention about providing further bitwise operation, which is disclosed by Kontio, e.g., paragraphs 54 and 55. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the well-known concept of providing further bitwise operation with the teachings of Aviani-Cisco-Technology, Khieu and Kontio in order to facilitate usage of further bitwise operation because it would enhance comparison of strings content performed with the bitwise operation. The compared information would be used for determining case insensitive match.

85. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Aviani-Cisco-Technology in view of Khieu and Slater et al., 6,654,796, Cisco (Hereinafter Slater).

86. Referring to claim 21, Aviani-Cisco-Technology and Khieu discloses the claimed limitations as rejected above. Aviani-Cisco-Technology and Khieu does not specifically mention about the network being WAN, which is disclosed by Slater, e.g., col., 1, lines 55 – 67, col., 9, lines 42 – 65. It would have been obvious to one of ordinary skill in the art at the time the

invention was made to include the well-known concept of providing the WAN with the teachings of Aviani-Cisco-Technology, Khieu and Slater in order to facilitate usage of the WAN because it would enhance receiving strings for comparison from different networks. The compared information of the strings from different networks would be used for determining case insensitive match.

87. Claims 1, 24-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wein et al. 7,240,100, Akami Technology (Hereinafter Wein-Akami-Technology) in view of Chopra et al., 6,631,466 (Hereinafter Chopra).

88. Referring to claim 1, Wein-Akami-Technology discloses a computer implemented method for comparing an unknown string to a predefined string (col., 5, line 10 - col., 6, line 59), the method comprising: storing, on a network device a database containing a plurality of predefined string (col., 5, line 10 - col., 6, line 59), wherein the predefined strings stored within the database represent known headers for a network communication protocol (col., 5, line 10 - col., 6, line 59); receiving, with the network device, a network message in response to receiving the network message (col., 5, line 10 - col., 6, line 59), selecting one of the plurality of predefined strings stored within the database of the network device (col., 5, line 10 - col., 6, line 59); identifying a portion of the network message as an unknown string for comparison with the selected predefined string ( col., 5, line 10 - col., 6, line 59); performing an operation between an ASCII binary representation of at least a segment of the unknown string and an ASCII binary representation of at least a segment of the selected predefined string ( col., 5, line 10 - col., 6,

line 59); to produce an indication for a case-insensitive string match wherein the indication for the case-insensitive string match indicates whether all characters of the unknown string within the network message match all corresponding characters of the selected predefined string so as to match one of the known headers of the network communication protocol (col., 5, line 10 - col., 6, line 59, http strings are case insensitive and http strings comparison contain operations); processing the network message based on the indication of a case insensitive string match (col., 5, line 10 - col., 6, line 59); and outputting a response from the network device based on the processed network message (col., 5, line 10 - col., 6, line 59).

Wein-Akami-Technology does not disclose performing a bitwise exclusive OR operation and performing a bitwise operation between a predefined flag and a result of the exclusive OR operation; and comparing the predefined flag and a result of the bitwise operation. Chopra discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., usage of HTTP header matching by performing bitwise operations including shifting left, etc., figure 7A and 7B and its description. However, since Wein-Akami-Technology does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Wein-Akami-Technology is open for different techniques for string comparisons, and the ability to be adaptive to enables Wein-Akami-Technology's system/method/medium to be implemented in a wider variety of techniques including Chopra's technique for string comparison.

89. Referring to claim 24, Wein-Akami-Technology discloses a method of case-insensitive string matching for use in a computer network (col., 5, line 10 - col., 6, line 59), the method

comprising: storing, on a network device, a plurality of predefined strings, wherein the predefined strings represent known headers for a network communication protocol (col., 5, line 10 - col., 6, line 59); receiving with the network device, the network message selecting one of the plurality of predefined strings stored within the network device (col., 5, line 10 - col., 6, line 59); identifying a portion of the network message as an unknown string for comparison with the selected predefined string (col., 5, line 10 - col., 6, line 59); to produce a single bit output that indicates whether a case-insensitive match exists between the selected predefined string and the unknown string (col., 5, line 10 - col., 6, line 59, http strings are case insensitive and http strings comparison contain operations); processing the network message based on the indication of the case-insensitive match (col., 5, line 10 - col., 6, line 59); and outputting a response from the network device (col., 5, line 10 - col., 6, line 59).

Wein-Akami-Technology does not disclose performing a bitwise exclusive OR operation and performing a bitwise OR operation between a results of the bitwise exclusive OR operation and a predetermined flag; and comparing the predetermined flag and a result of the bitwise OR operation. Chopra discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., usage of HTTP header matching by performing bitwise operations including shifting left, etc., figure 7A and 7B and its description. However, since Wein-Akami-Technology does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Wein-Akami-Technology is open for different techniques for string comparisons, and the ability to be adaptive to enables Wein-Akami-Technology's system/method/medium to be implemented in a wider variety of techniques including Chopra's technique for string comparison.

90. Referring to claim 25, Wein-Akami-Technology discloses a computer networking device for improving data transfer via a computer network, the device comprising a processor configured to compare a client HTTP header with a known HTTP header by storing, on the networking device, a database containing a plurality of known HTTP headers(col., 5, line 10 - col., 6, line 59); receiving with the networking device, a client HTTP header in response to receiving the client HTTP header, selecting one of the known HTTP headers stored within the database of the network device (col., 5, line 10 - col., 6, line 59); to produce an indication for a case-insensitive string match between the client HTTP header (col., 5, line 10 - col., 6, line 59, http strings are case insensitive and http strings comparison contain operations) and the selected known HTTP header predefined processing the network message based on the indication of the case-insensitive match and outputting a response from the network device based on the processed network message (col., 5, line 10 - col., 6, line 59).

Wein-Akami-Technology does not disclose performing a bitwise exclusive OR operation and performing a bitwise OR operation between a result of the exclusive OR, operation and a predetermined flag; and comparing the predetermined flag and a result of the bitwise OR operation. Chopra discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., usage of HTTP header matching by performing bitwise operations including shifting left, etc., figure 7A and 7B and its description. However, since Wein-Akami-Technology does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Wein-Akami-Technology is open for different techniques for string comparisons, and the ability to be adaptive to enables

Wein-Akami-Technology's system/method/medium to be implemented in a wider variety of techniques including Chopra's technique for string comparison.

91. Referring to claim 26, Wein-Akami-Technology an article of manufacture comprising a storage medium having a plurality of machine-readable instructions, wherein when the instructions arc executed by a computing system, the instructions providing for: storing, on a network device, a database containing a plurality of predefined strings (col., 5, line 10 - col., 6, line 59), wherein the predefined strings stored, within the database represent known headers for a network communication protocol receiving with the network device (col., 5, line 10 - col., 6, line 59), a network message in response to receiving the network message (col., 5, line 10 - col., 6, line 59), selecting one of the plurality of predefined strings, stored within the database of the network device (col., 5, line 10 - col., 6, line 59); identifying a portion of the network message as an unknown string for comparison with the selected predefined string (col., 5, line 10 - col., 6, line 59); performing an operation between an ASCII binary representation of at least a segment of the unknown string and an ASCII binary representation of at least a segment of the selected predefined string ( col., 5, line 10 - col., 6, line 59); to produce an indication for a case-insensitive string match between the predefined string and the unknown string (col., 5, line 10 - col., 6, line 59, http strings are case insensitive and http strings comparison contain operations), wherein the indication for the case-insensitive match indicates whether all characters of the unknown string within the network message match all corresponding characters of the identified predefined string so as to match one of the known headers of the network communication protocol (col., 5, line 10 - col., 6, line 59); processing the network message based on the

indication of the case-insensitive match and outputting a response from the network device based on the processed network message (col., 5, line 10 - col., 6, line 59).

Wein-Akami-Technology does not disclose performing a bitwise exclusive OR operation and performing a bitwise operation between a predefined flag and a result of the exclusive OR operation; and comparing the predefined flag and a result of the bitwise operation. Chopra discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., usage of HTTP header matching by performing bitwise operations including shifting left, etc., figure 7A and 7B and its description. However, since Wein-Akami-Technology does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Wein-Akami-Technology is open for different techniques for string comparisons, and the ability to be adaptive to enables Wein-Akami-Technology's system/method/medium to be implemented in a wider variety of techniques including Chopra's technique for string comparison.

92. Claims 1, 24-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Giles et al. 6,986,047, International Business Machines (Hereinafter Giles-IBM) in view of view of James et al., 6,523,108 (Hereinafter James).

93. Referring to claim 1, Giles-IBM discloses a computer implemented method for comparing an unknown string to a predefined string (comparison of http header hash, col., 5, line 12 - col., 6, line 58), the method comprising: storing, on a network device a database containing a plurality of predefined string (col., 5, line 12 - col., 6, line 58), wherein the predefined strings stored within the database represent known headers for a network communication protocol (col.,

5, line 12 - col., 6, line 58); receiving, with the network device, a network message in response to receiving the network message (col., 5, line 12 - col., 6, line 58), selecting one of the plurality of predefined strings stored within the database of the network device (col., 5, line 12 - col., 6, line 58); identifying a portion of the network message as an unknown string for comparison with the selected predefined string ( col., 5, line 12 - col., 6, line 58); performing an operation between an ASCII binary representation of at least a segment of the unknown string and an ASCII binary representation of at least a segment of the selected predefined string ( col., 5, line 10 - col., 6, line 58); to produce an indication for a case-insensitive string match wherein the indication for the case-insensitive string match indicates whether all characters of the unknown string within the network message match all corresponding characters of the selected predefined string so as to match one of the known headers of the network communication protocol (col., 5, line 12 - col., 6, line 58); processing the network message based on the indication of a case insensitive string match (col., 5, line 12 - col., 6, line 58); and outputting a response from the network device based on the processed network message (response messages based on comparison of http header hash, col., 5, line 12 - col., 6, line 58).

Giles-IBM does not disclose performing a bitwise exclusive OR operation and performing a bitwise operation between a predefined flag and a result of the exclusive OR operation; and comparing the predefined flag and a result of the bitwise operation. James discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., usage of XOR, etc., figures 9A-9D and its description. However, since Giles-IBM does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Giles-IBM is open for different techniques for

string comparisons, and the ability to be adaptive to enables Giles-IBM's system/method/medium to be implemented in a wider variety of techniques including James's technique for string comparison.

94. Referring to claim 24, Giles-IBM discloses a method of case-insensitive string matching for use in a computer network (col., 5, line 12 - col., 6, line 58), the method comprising: storing, on a network device, a plurality of predefined strings, wherein the predefined strings represent known headers for a network communication protocol (col., 5, line 12 - col., 6, line 58); receiving with the network device, the network message selecting one of the plurality of predefined strings stored within the network device (col., 5, line 12 - col., 6, line 58); identifying a portion of the network message as an unknown string for comparison with the selected predefined string (col., 5, line 12 - col., 6, line 58); to produce a single bit output that indicates whether a case-insensitive match exists between the selected predefined string and the unknown string (col., 5, line 12 - col., 6, line 58, http strings are case insensitive and http strings comparison contain operations); processing the network message based on the indication of the case-insensitive match (col., 5, line 12 - col., 6, line 58); and outputting a response from the network device (col., 5, line 12 - col., 6, line 58).

Giles-IBM does not disclose performing a bitwise exclusive OR operation and performing a bitwise OR operation between a results of the bitwise exclusive OR operation and a predetermined flag; and comparing the predetermined flag and a result of the bitwise OR operation to produce a single bit output. James discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common

case, e.g., usage of XOR, etc., figures 9A-9D and its description. However, since Giles-IBM does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Giles-IBM is open for different techniques for string comparisons, and the ability to be adaptive to enables Giles-IBM's system/method/medium to be implemented in a wider variety of techniques including James's technique for string comparison.

95. Referring to claim 25, Giles-IBM discloses a computer networking device for improving data transfer via a computer network, the device comprising a processor configured to compare a client HTTP header with a known HTTP header by storing, on the networking device, a database containing a plurality of known HTTP headers(col., 5, line 12 - col., 6, line 58); receiving with the networking device, a client HTTP header in response to receiving the client HTTP header, selecting one of the known HTTP headers stored within the database of the network device (col., 5, line 12 - col., 6, line 58); to produce an indication for a case-insensitive string match between the client HTTP header (col., 5, line 12 - col., 6, line 58, http strings are case insensitive and http strings comparison contain operations) and the selected known HTTP header predefined processing the network message based on the indication of the case-insensitive match and outputting a response from the network device based on the processed network message (col., 5, line 12 - col., 6, line 58).

Giles-IBM does not disclose performing a bitwise exclusive OR operation and performing a bitwise OR operation between a result of the exclusive OR, operation and a predetermined flag; and comparing the predetermined flag and a result of the bitwise OR operation. James discloses these limitations and well known techniques of comparing case-

insensitive strings without converting the strings to a common case, e.g., usage of XOR, etc., figures 9A-9D and its description. However, since Giles-IBM does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Giles-IBM is open for different techniques for string comparisons, and the ability to be adaptive to enables Giles-IBM's system/method/medium to be implemented in a wider variety of techniques including James's technique for string comparison.

96. Referring to claim 26, Giles-IBM an article of manufacture comprising a storage medium having a plurality of machine-readable instructions, wherein when the instructions arc executed by a computing system, the instructions providing for: storing, on a network device, a database containing a plurality of predefined strings (col., 5, line 12 - col., 6, line 58), wherein the predefined strings stored, within the database represent known headers for a network communication protocol receiving with the network device (col., 5, line 12 - col., 6, line 58), a network message in response to receiving the network message (col., 5, line 12 - col., 6, line 58), selecting one of the plurality of predefined strings, stored within the database of the network device (col., 5, line 12 - col., 6, line 58); identifying a portion of the network message as an unknown string for comparison with the selected predefined string (col., 5, line 12 - col., 6, line 58); performing an operation between an ASCII binary representation of at least a segment of the unknown string and an ASCII binary representation of at least a segment of the selected predefined string ( col., 5, line 10 - col., 6, line 58) to produce an indication for a case-insensitive string match between the predefined string and the unknown string (col., 5, line 12 - col., 6, line 58, http strings are case insensitive and http strings comparison contain operations), wherein the

indication for the case-insensitive match indicates whether all characters of the unknown string within the network message match all corresponding characters of the identified predefined string so as to match one of the known headers of the network communication protocol (col., 5, line 12 - col., 6, line 58); processing the network message based on the indication of the case-insensitive match and outputting a response from the network device based on the processed network message (col., 5, line 12 - col., 6, line 58).

Giles-IBM does not disclose performing a bitwise exclusive OR operation and performing a bitwise operation between a predefined flag and a result of the exclusive OR operation; and comparing the predefined flag and a result of the bitwise operation. James discloses these limitations and well known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., usage of XOR, etc., figures 9A-9D and its description. However, since Giles-IBM does not limit its string comparison to certain technique, it is obvious to one of ordinary skill in the art that Giles-IBM is open for different techniques for string comparisons, and the ability to be adaptive to enables Giles-IBM's system/method/medium to be implemented in a wider variety of techniques including James's technique for string comparison.

***Conclusion***

Examiner has cited particular columns and/or paragraphs and/or sections and/or page numbers in the reference(s) as applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to the specific limitations within the individual claim, other passages and figures may apply as well.

It is respectfully requested from the applicant in preparing responses, to fully consider the references in entirety, as potentially teaching, all or part of the claimed invention, as well as the context of the passage, as taught by the prior art or disclosed by the Examiner.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Haresh Patel whose telephone number is (571) 272-3973. The examiner can normally be reached on Monday, Tuesday, Thursday and Friday from 10:00 am to 8:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nathan Flynn can be reached at (571) 272-1915. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Haresh N. Patel/

Primary Examiner, Art Unit 2454

3/8/10